Федеральное государственное бюджетное образовательное учреждение высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации».

## Кафедра «Прикладная Информатика»

ОТЧЕТ

О ПРОДЕЛАННОЙ ПРАКТИЧЕСКОЙ РАБОТЕ №1

по курсу «Базы данных»

Выполнил: студент группы

ИК-721

**(наименование группы)**

***Соколов Дмитрий Александрович***

**(Ф.И.О.)**

## Нижний Новгород

**2024 г.**

Предметная область: «Автомобили»

**Шаги проектирования:**

1. Для предметной области разработать набор атрибутов универсальной таблицы.

|  |  |
| --- | --- |
| **Атрибут** | **Описание** |
| car\_id | Уникальный идентификатор автомобиля |
| car\_series | Серия автомобиля |
| manufacturer | Завод-изготовитель |
| engine\_power | Мощность двигателя (л.с.) |
| passenger\_count | Число пассажиров |
| weight | Вес автомобиля (кг) |
| color | Цвет автомобиля |
| body\_type | Тип кузова |
| air\_conditioning | Наличие кондиционера (Да/Нет) |
| service\_center\_id | Уникальный идентификатор сервисного центра |
| service\_center\_name | Наименование сервисного центра |
| company\_name | Наименование фирмы (владельца сервисного центра) |
| repair\_slots | Число мест для ремонта и обслуживания |
| service\_center\_address | Адрес сервисного центра |

1. Выявить набор функциональных и многозначных зависимостей между атрибутами.

### Основные функциональные зависимости:

1. car\_id → car\_series, manufacturer, engine\_power, passenger\_count, weight, color, body\_type, air\_conditioning
   * Уникальный идентификатор автомобиля определяет его характеристики.
2. service\_center\_id → service\_center\_name, company\_name, repair\_slots, service\_center\_address
   * Уникальный идентификатор сервисного центра определяет его свойства.
3. car\_id → service\_center\_id
   * Каждый автомобиль обслуживается в одном конкретном сервисном центре.
4. car\_series → manufacturer
   * Серия автомобиля определяет его завод-изготовитель.
5. Используя алгоритм Бернштейна разработать набор таблиц базы данных. Проверить полученные таблицы на соответствие третьей и четвертой нормальным формам.

### **Шаг 3.1. Декомпозиция на 1NF**

Все атрибуты уже атомарны, т.е. таблица находится в 1NF.

### **Шаг 3.2. Декомпозиция на 2NF**

Таблица должна быть декомпозирована для удаления частичных зависимостей:

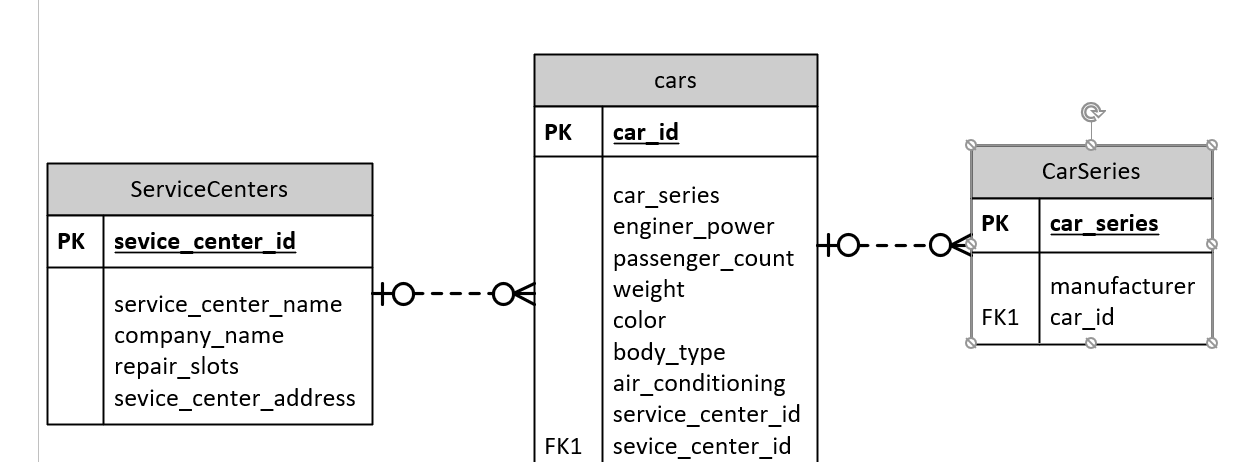
* Таблица **Cars**:  
  Определяется зависимость car\_id → car\_series, manufacturer, engine\_power, passenger\_count, weight, color, body\_type, air\_conditioning.
* Таблица **ServiceCenters**:  
  Определяется зависимость service\_center\_id → service\_center\_name, company\_name, repair\_slots, service\_center\_address.

### **Шаг 3.3. Декомпозиция на 3NF**

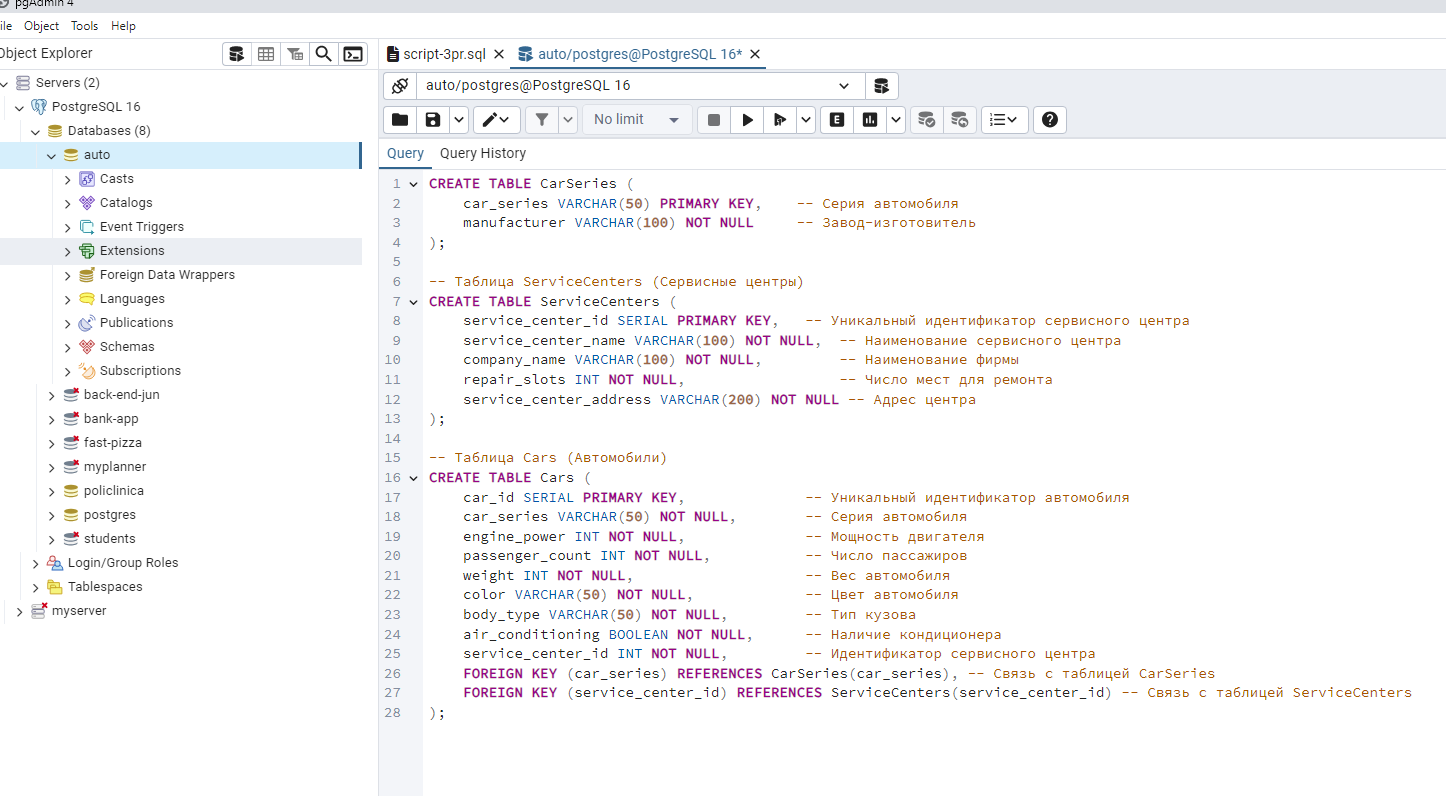
Для устранения транзитивных зависимостей:

* Создадим отдельную таблицу **CarSeries**:  
  car\_series → manufacturer

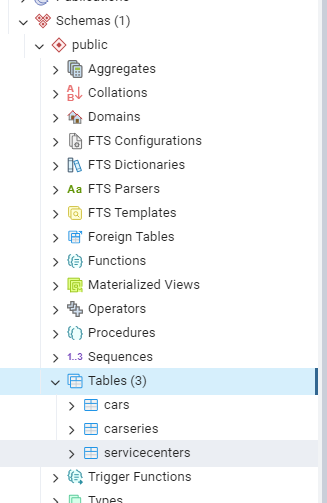
1. Установить связи между таблицами и построить реляционную модель базы данных.



1. Получить схему базы данных для выбранной СУБД и сформировать команды создания таблиц и индексов.



1. Выполнить сгенерированные команды SQL для формирования таблиц и индексов БД.

****